FIGURES

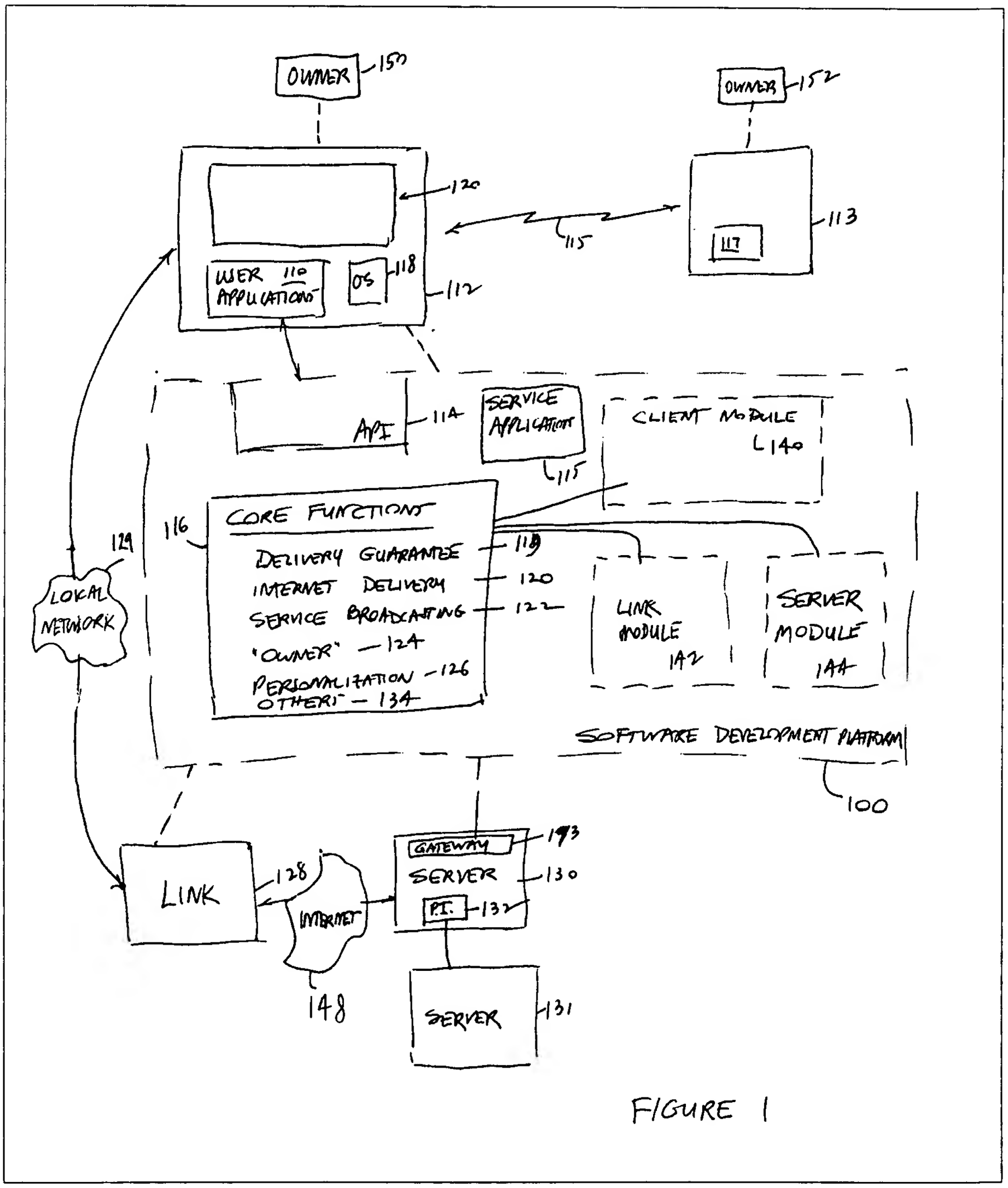

FIGURE 1

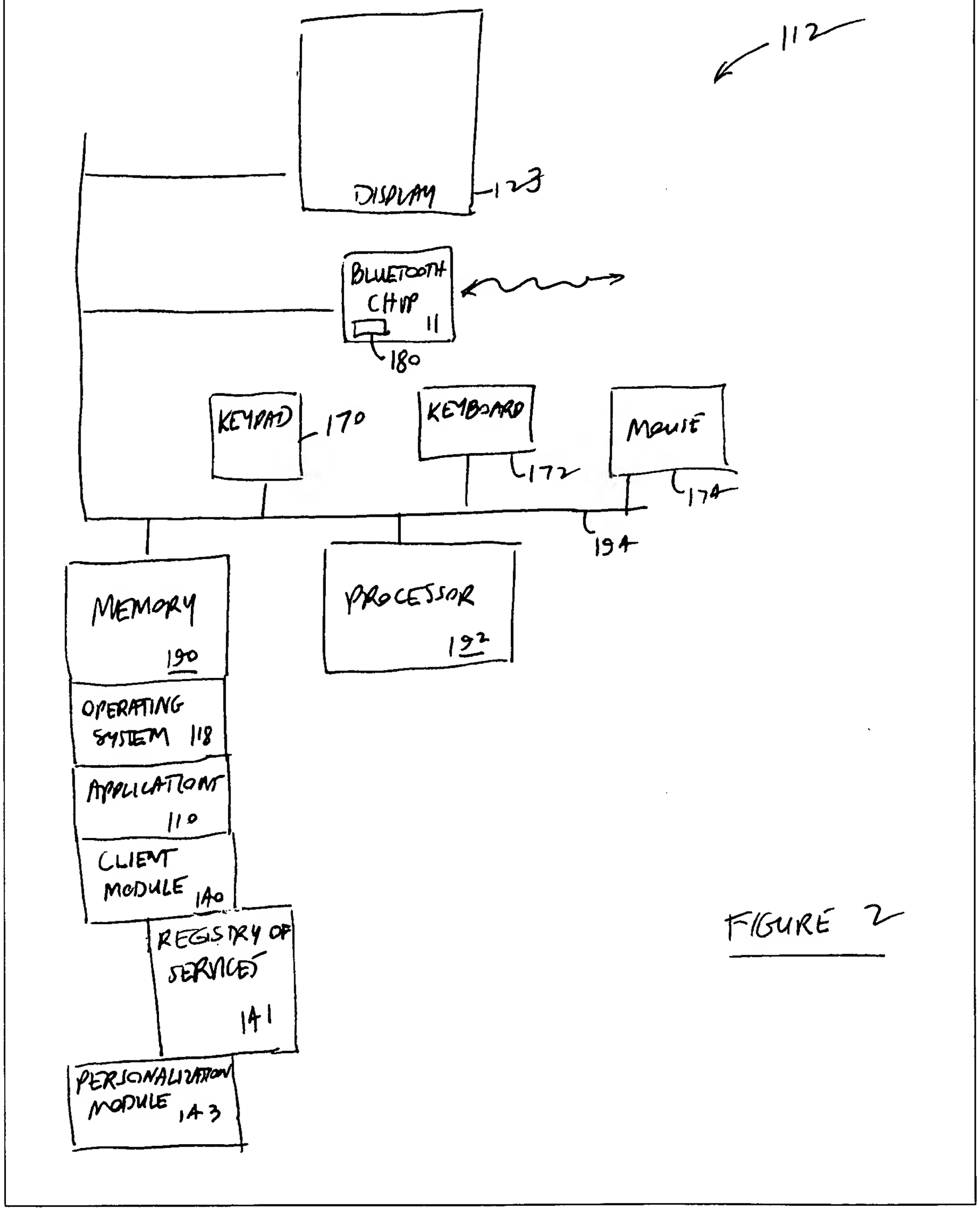
112
DISPLAY
123
BLUETOOTH CHIP
11
180
KEYPAD
170
KEYBOARD
172
MOUSE
174
194
MEMORY
190
PROCESSOR
192
OPERATING SYSTEM 118
APPLICATION(S) 110
CLIENT MODULE 120
REGISTRY OF SERVICES
141
PERSONALIZATION MODULE 143

FIGURE 2

```
• LocalService {
    char * serviceName;
    int receiveWhenRunning;
    int receiveWhenNotRunning;
    int running;
    int maxNumOfMessagesToStore;
    char * exp_fld1;
    char * exp_fld2;
    char * exp_fld3;
    char * exp_fld4;
    char * exp_fld5;
  };
```

FIG. 3A

```
• RemoteService {
    char * serviceName;
    char * userName;
  };
```

FIG. 3B

```
• KMessage {
    char * serviceName;
    char * date;
    char * recipient;
    char * sender;
    char * messageBody;
  };
```

FIG. 3C

FIG. 4A

Name	sendMessage
Arguments	char * to, char * serviceName, char * data
Return Values	int err
Description	This function provides sending capabilities so that messages or any kind of unformatted text can be sent between Bluetooth devices. Reception of the text is guaranteed, because even when the devices are not within range, the text is stored and communicated via an Internet connection. If a user is logged in to more than one device simultaneously, the message/text will be sent to both devices at the same time.

FIG. 4B

Name	getMessages
Arguments	struct KMessage * message
Return Values	int err
Description	The getMessages function retrieves all messages or any other unformatted text sent from another Bluetooth device. It returns the data in the KMessage data structure. If a user is logged in to more than one device simultaneously, the message/text will be received from both devices at the same time.

FIG. 4C

Name	getMessage
Arguments	struct KMessage * message
Return Values	int err
Description	The getMessage function retrieves just one message or any other unformatted text sent from another Bluetooth device. It returns the data in the KMessage data structure. If a user is logged in to more than one device simultaneously, the message/text will be received from both devices at the same time.

FIG. 4D

Name	getSurroundingServices
Arguments	struct RemoteService areaServices []
Return Values	int err
Description	This function returns an array of mappings of users and services available on that user's device. This information was previously stored in a database termed the registry, which is a list of devices within range of a Bluetooth device.

FIG. 4E

Name	AddService
Arguments	char * serviceName
Return Values	int err
Description	This function adds a service entry to the registry.

FIG. 4F

Name	RemoveService
Arguments	char * serviceName
Return Values	int err
Description	This function removes a service entry from the registry.

FIG. 4G

Name	changePMTdata
Arguments	<<waiting to hear about PMT API>>
Return Values	int err
Description	A function that allows users to update their personal PMT data and preferences using their particular devices. If the device is not within Bluetooth range of an Internet connection, it will store these update preferences, and make changes within the permanent PMT upon coming into contact with an Internet connection.

FIG. 4H

Name	GetPMTdata
Arguments	char * user
Return Values	int err
Description	Allows a service to get the PMT data of a particular user from the PMT database. If the service cannot reach the PMT database, the information comes from the local storage on the device of the user. Only information that is designated as shared or public data will be retrieved.

FIG. 4I

Name	ChangePMTpermissions
Arguments	<<waiting to hear about PMT API>>
Return Values	int err
Description	This function allows a user to change his PMT permissions from his device.

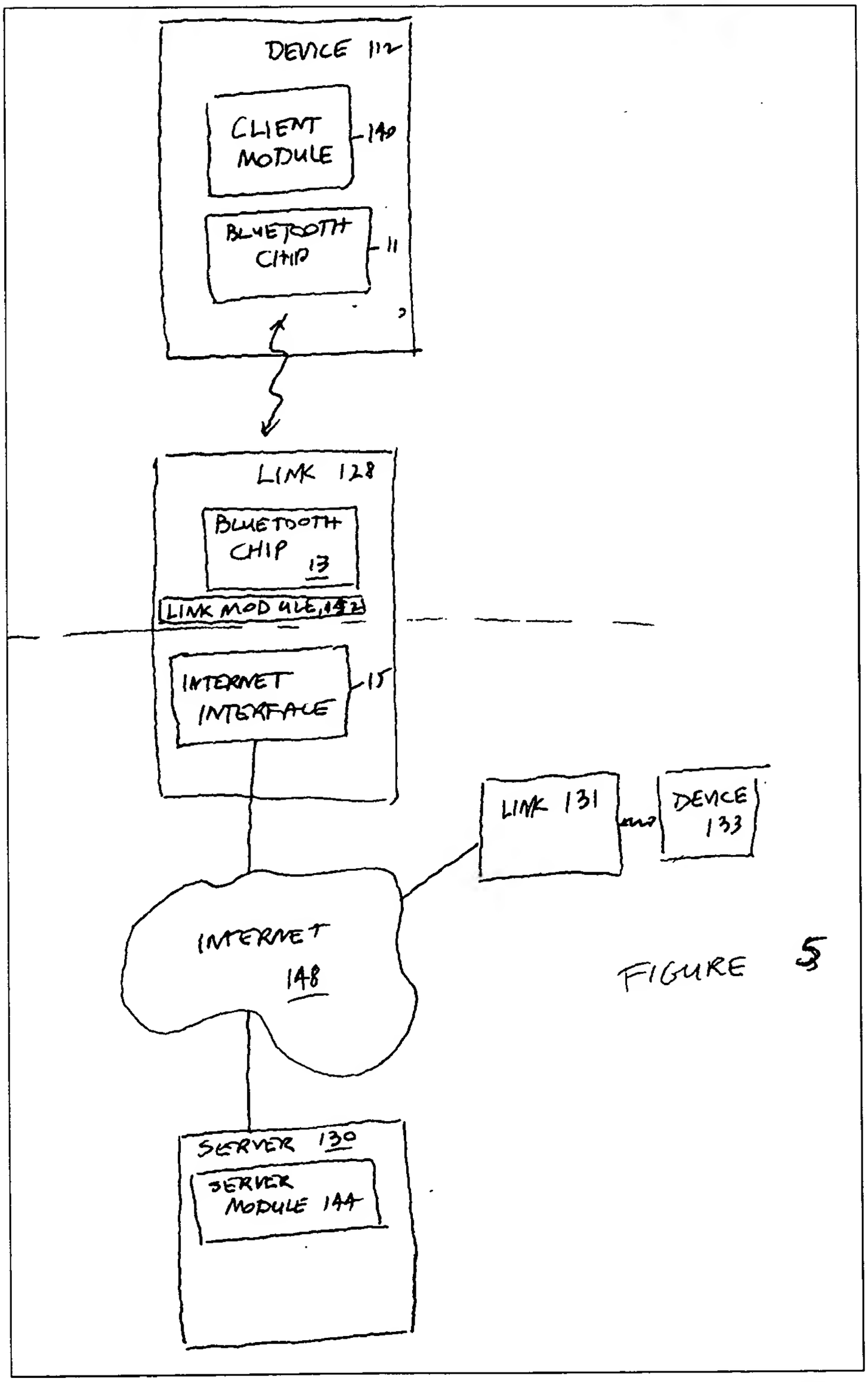
DEVICE 112
CLIENT MODULE
140
BLUETOOTH CHIP
11
LINK 128
BLUETOOTH CHIP 13
LINK MODULE 142
INTERNET INTERFACE
15
LINK 131
DEVICE 133
INTERNET 148
SERVER 130
SERVER MODULE 144

FIGURE 5

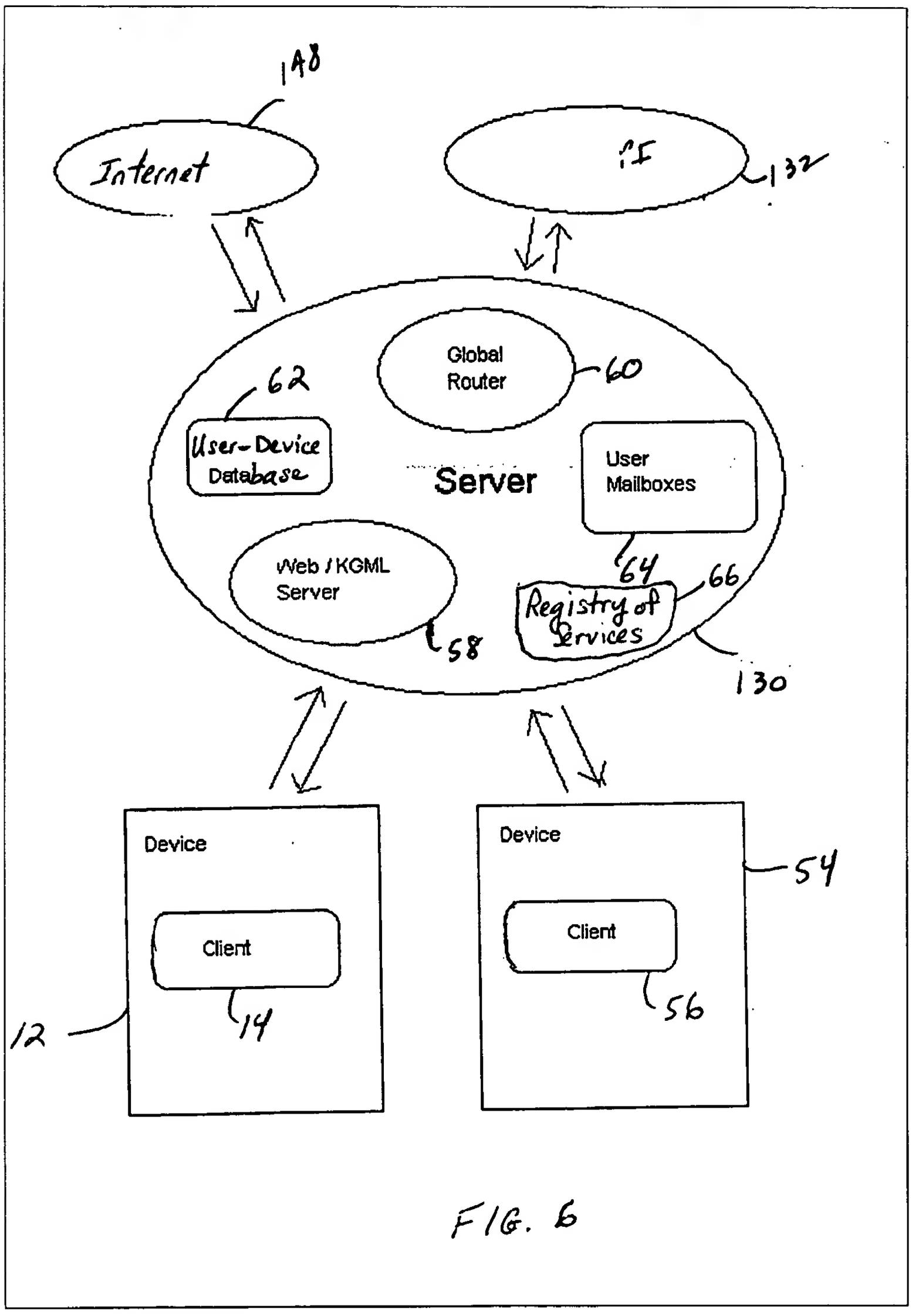
48
Internet
IP
132
Global Router
60
62
User-Device Database
Server
User Mailboxes
64
66
Web / KGML Server
58
Registry of Services
130
Device
Client
12
14
Device
54
Client
56

FIG. 6

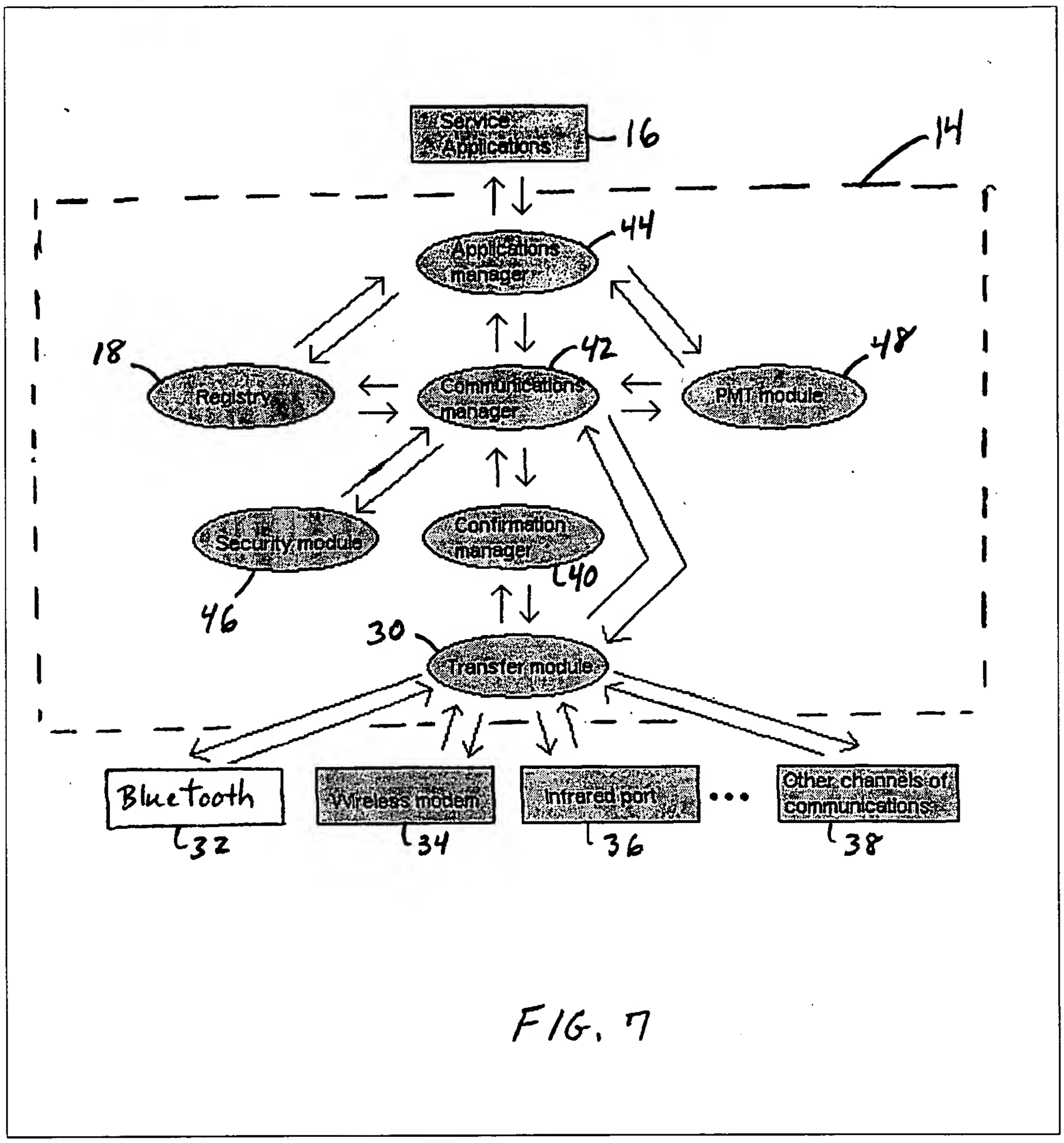
Service Applications
16
14
Applications manager
44
18
Registry
Communications manager
42
PMT module
48
Security module
46
Confirmation manager
40
30
Transfer module
Bluetooth
32
Wireless modem
34
Infrared port
36
Other channels of communications
38

FIG. 7

100 The particular service application passes a message with an identification of the intended recipient to the applications manager

↓

102 The applications manager passes the message and recipient's identification to the communications manager and requests that the message be transmitted in a secure and guaranteed manner

↓

104 The communications manager passes the message and security information to the security module

↓

106 The security module executes the cryptographic algorithm on the message and returns the encrypted message to the communications manager

↓

108 The communications manager passes the encrypted message to the confirmation manager and specifies the channel of communication

↓

110 The confirmation manager generates a message identification number and attaches it to the message header

↓

112 The confirmation manager adds the encrypted message to its queue of non-confirmed messages

↓

114 The confirmation manager passes the encrypted message and the specified channel to the transfer module

↓

116 The transfer module sends the encrypted message over the specified communication channel

↓

118 The transfer module in the recipient device receives the encrypted message and passes it to the recipient device's confirmation manager

↓

120 The confirmation manager in the recipient device passes the message to the communications manager

122 The confirmation manager in the recipient device adds the message identification number to its list of received messages and generates a confirmation message that is passed to the transfer module in the recipient device

↓

TO 124

130 The communications manager in the recipient device removes the header

↓

TO 132

FIG. 8A

124 The transfer module sends the confirmation message over the same communication channel over which the original message arrived

↓

126 The transfer module in the sending device receives the confirmation message and passes it to the confirmation manager

↓

128 The confirmation manager removes the original message from its queue of messages that are awaiting confirmation

132 The communication manager passes the encrypted, digitally signed message to the security module in the recipient device

↓

134 The security module responds with the sender's identity

↓

136 The communications manager in the recipient device adds the decrypted message to its message queue

↓

138 The communications manager notifies the user and service application according to the settings in the registry of the recipient device

↓

140 The service application requests the message from the communications manager through the applications manager in the recipient device

↓

142 The communications manager passes the next message marked for the intended application from the message queue to the service application

↓

144 The communications manager deletes the message from its message queue

FIG. 8B

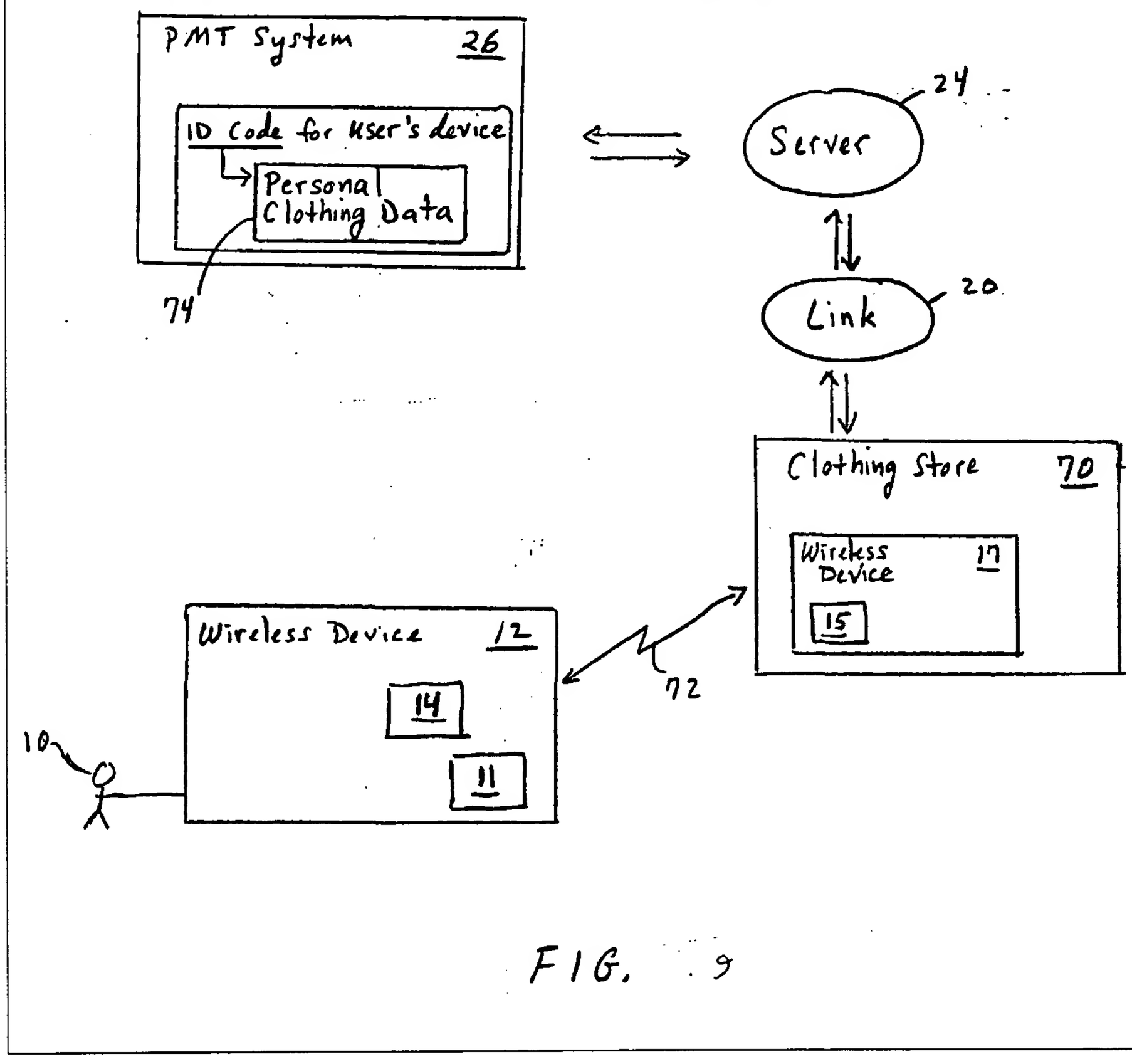
PMT System 26
ID Code for User's device
Persona Clothing Data
74
24
Server
Link
20
Clothing Store 70
Wireless Device 17
15
Wireless Device 12
14
11
10
72

FIG. 9

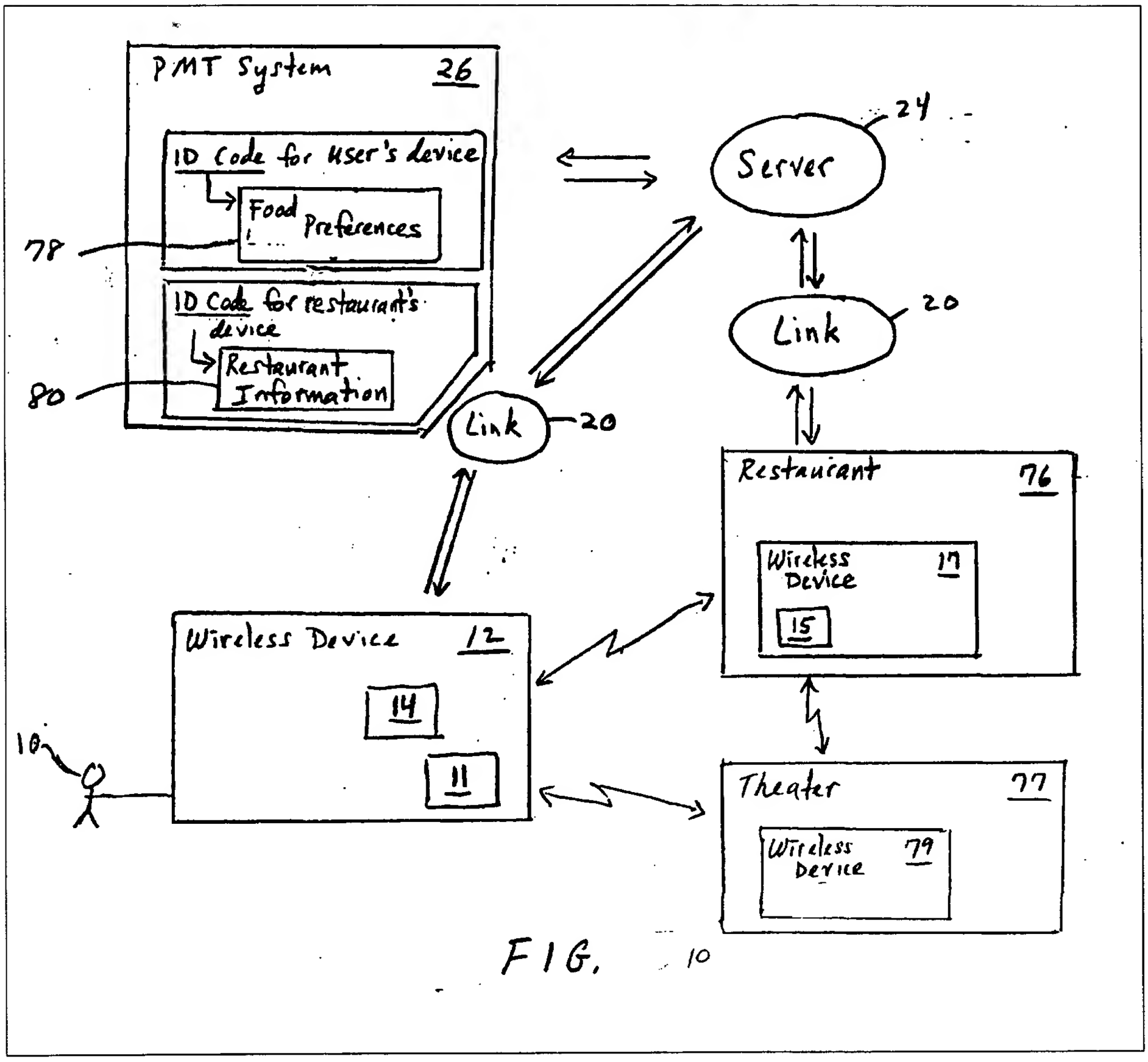
PMT System 26
ID Code for User's device
Food Preferences
78
ID Code for restaurant's device
Restaurant Information
80
Link
20
Server
24
Link
20
Restaurant
76
Wireless Device
17
15
Wireless Device
12
14
11
10
Theater
77
Wireless Device
79

FIG. 10

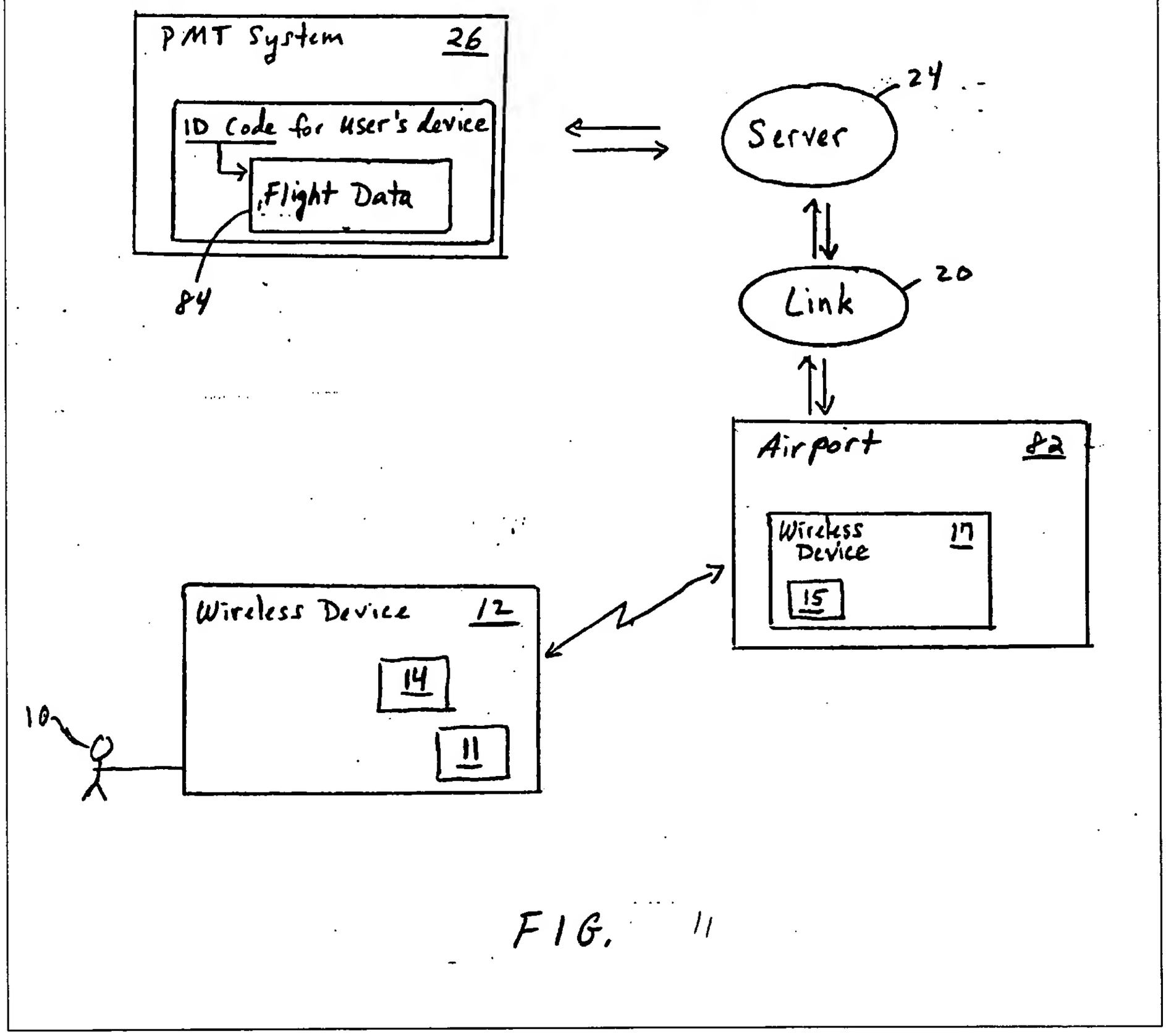
PMT System 26
ID Code for User's device
Flight Data
84
24
Server
20
Link
Airport 22
Wireless Device 17
15
Wireless Device 12
14
11
10

FIG. 11

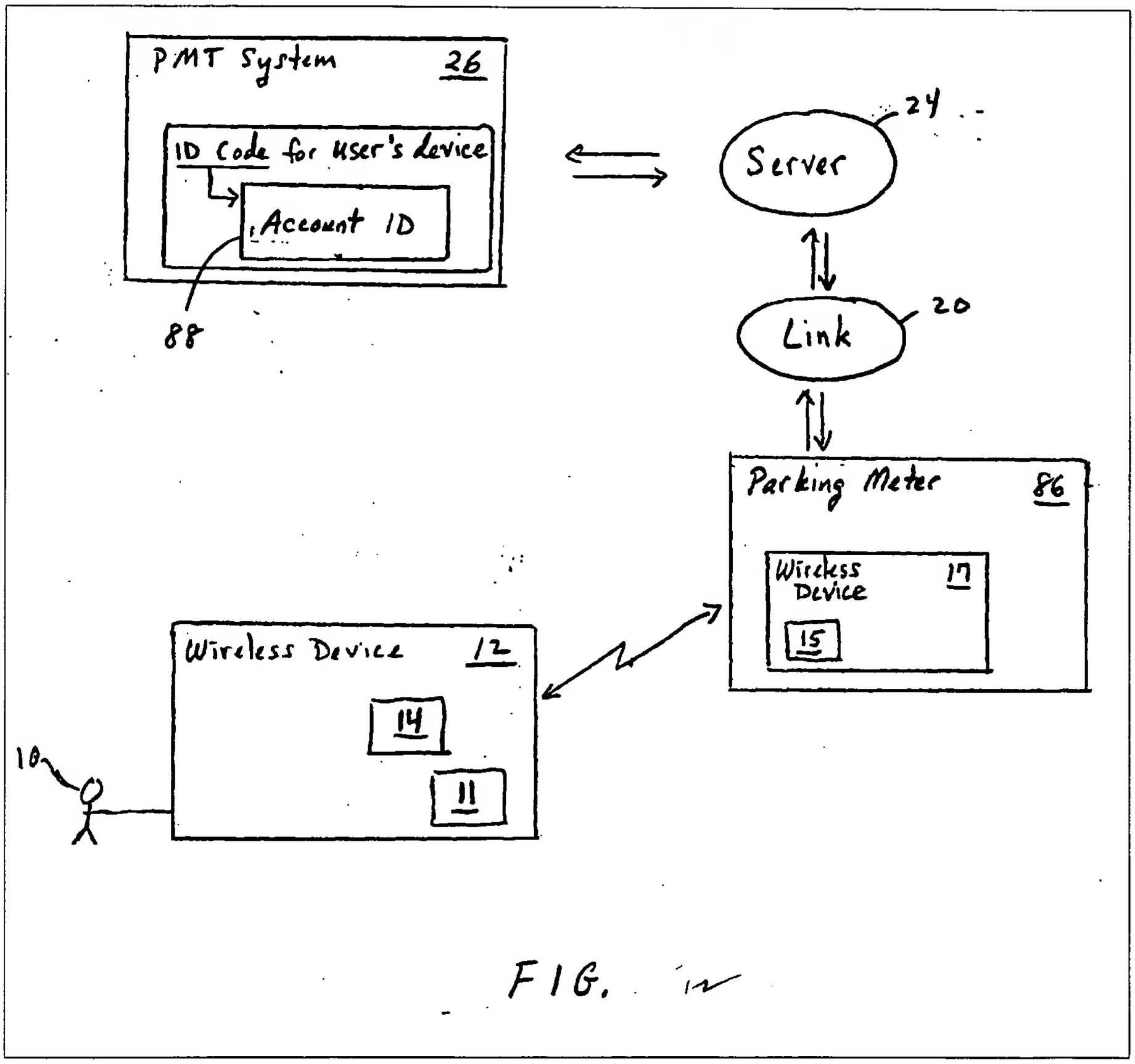
PMT System 26
ID Code for User's device
Account ID
88
24
Server
20
Link
Parking Meter 86
Wireless Device 17
15
Wireless Device 12
14
11
10

FIG. 1

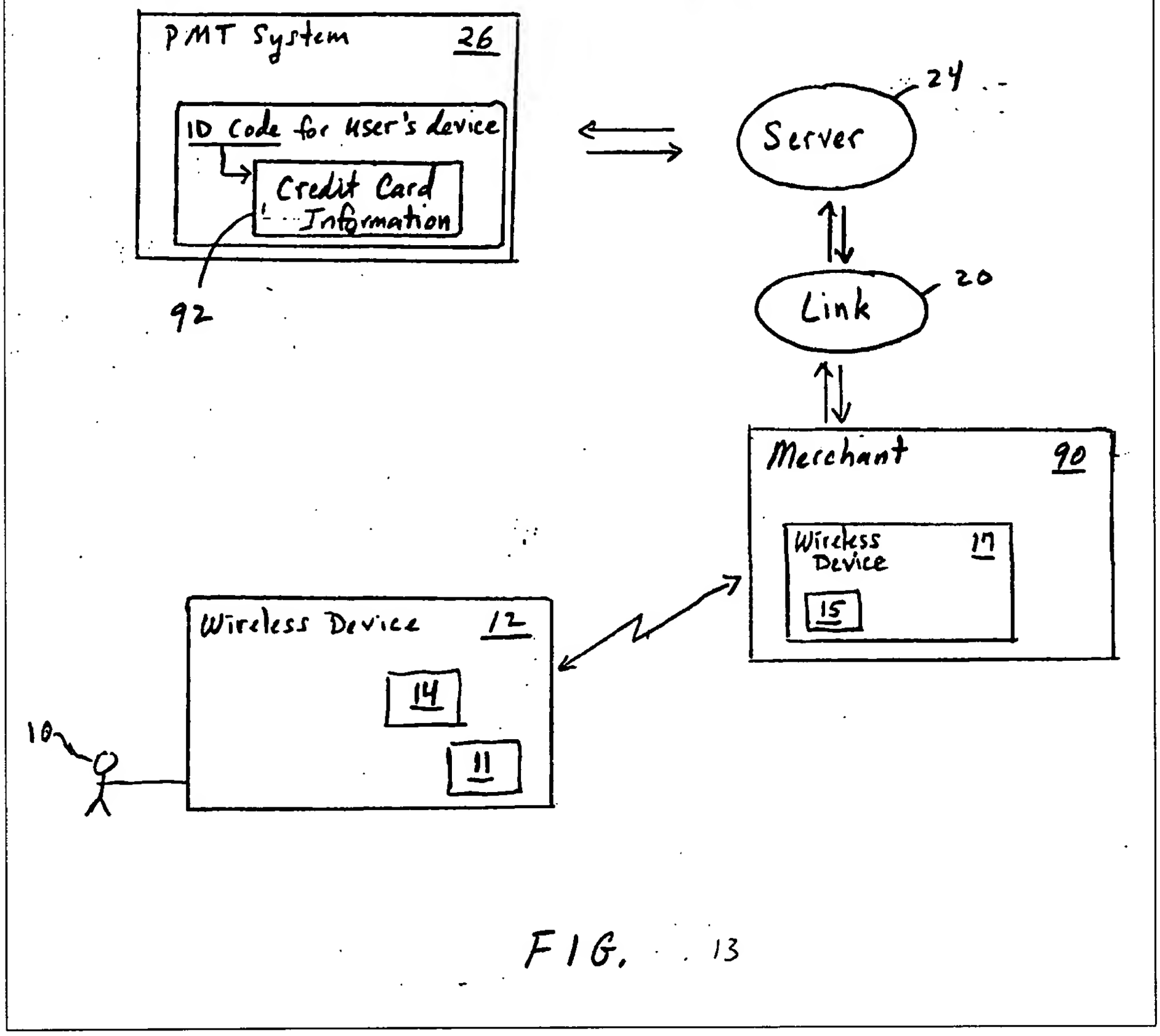
PMT System 26
ID Code for User's device
Credit Card Information
92
24
Server
Link
20
Merchant 90
Wireless Device 17
15
Wireless Device 12
14
11
10

FIG. 13

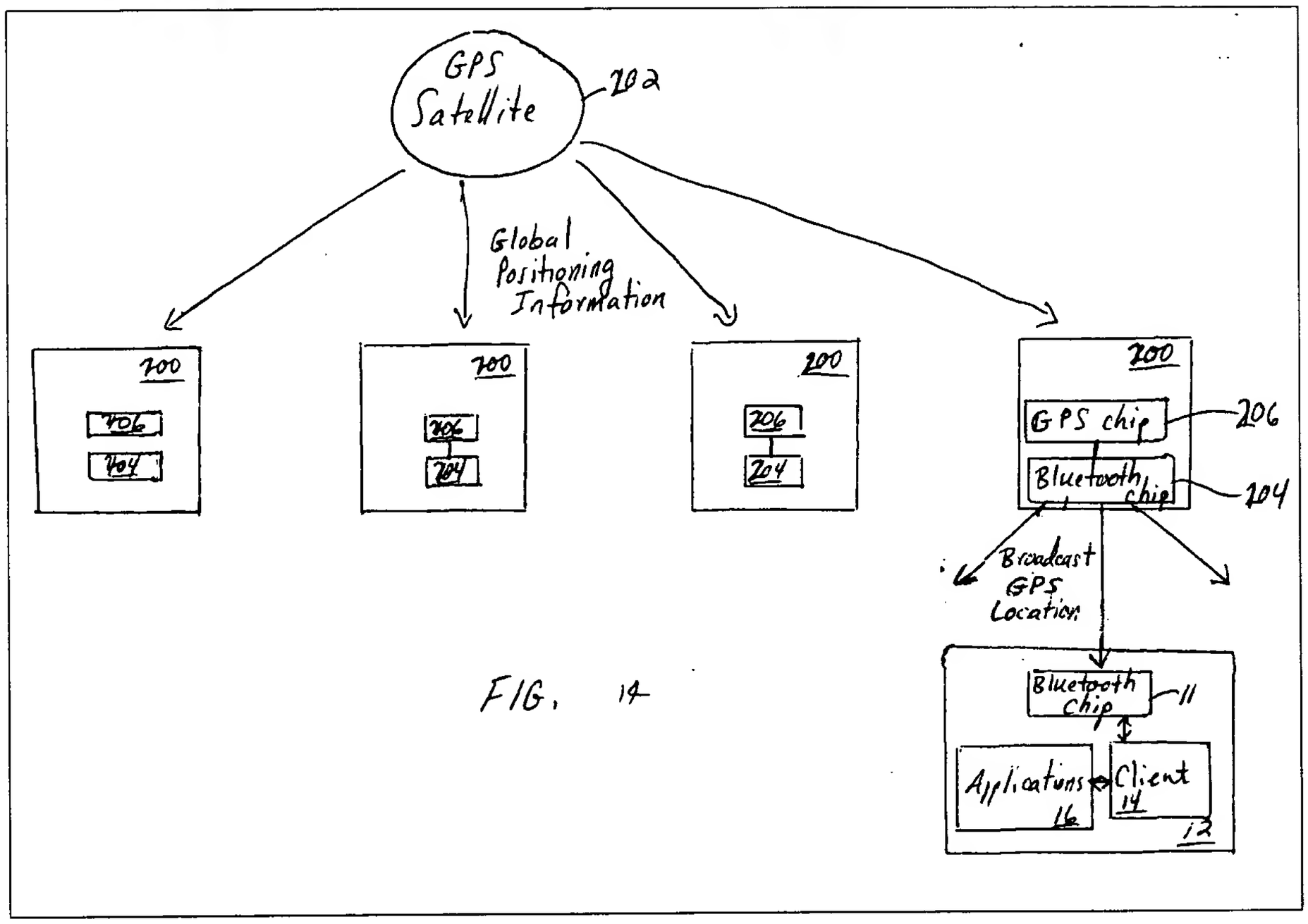
GPS Satellite
202
Global Positioning Information
200
206
204
GPS chip
Bluetooth chip
Broadcast GPS Location
Bluetooth Chip
11
Applications
16
Client
14
12

FIG. 14